Information and Software Technology xxx (2014) xxx–xxx

Contents lists available at ScienceDirect

Information and Software Technology

journal homepage: www.elsevier.com/locate/infsof

Current state of research on cross-site scripting (XSS) – A systematic literature review

Isatou Hydara *, Abu Bakar Md. Sultan, Hazura Zulzalil, Novia Admodisastro

Department of Software Engineering and Information System, Faculty of Computer Science and Information Technology, Universiti Putra Malaysia, 43400 Serdang, Selangor, Malaysia

ARTICLE INFO

Article history:
Received 9 December 2013
Received in revised form 8 May 2014
Accepted 11 July 2014
Available online xxxx

Keywords:
Systematic literature review
Cross-site scripting
Security
Web applications

ABSTRACT

Context: Cross-site scripting (XSS) is a security vulnerability that affects web applications. It occurs due to improper or lack of sanitization of user inputs. The security vulnerability caused many problems for users and server applications.
Objective: To conduct a systematic literature review on the studies done on XSS vulnerabilities and attacks.
Method: We followed the standard guidelines for systematic literature review as documented by Barbara Kitchenham and reviewed a total of 115 studies related to cross-site scripting from various journals and conference proceedings.
Results: Research on XSS is still very active with publications across many conference proceedings and journals. Attack prevention and vulnerability detection are the areas focused on by most of the studies. Dynamic analysis techniques form the majority among the solutions proposed by the various studies. The type of XSS addressed the most is reflected XSS.
Conclusion: XSS still remains a big problem for web applications, despite the bulk of solutions provided so far. There is no single solution that can effectively mitigate XSS attacks. More research is needed in the area of vulnerability removal from the source code of the applications before deployment.

Contents

* Corresponding author. Tel.: +60 176787278, +60 389466555.
E-mail addresses: ishahydara@gmail.com (I. Hydara), abakar@upm.edu.my (A.B.M. Sultan), hazura@upm.edu.my (H. Zulzalil), novia@upm.edu.my (N. Admodisastro).

http://dx.doi.org/10.1016/j.infsof.2014.07.010
0950-5849/© 2014 Elsevier B.V. All rights reserved.

1. Introduction

Accessing web applications has become a daily routine for many people. We depend on these applications to accomplish transactions, be it business, personal or otherwise. We interact dynamically with web applications when we access our emails, conduct banking transactions, visit social networking sites, etc. This dynamic nature of the web applications allows users to input information that will determine how a web site responds to the user. In many web sites, these user inputs are not properly validated thus making such a site vulnerable to cross-site scripting (XSS).

Cross-site scripting vulnerabilities (XSS henceforth) are a security problem that occurs in web applications. They were discovered in the 1990s in the early days of the World Wide Web [1]. They are among the most common and most serious security problems affecting web applications [2,3]. They are a type of injection problems [3] that enable malicious scripts to be injected into trusted web sites. This is a result of a failure to validate input from the web site users. What happens is either the web site fails to neutralize the user input or it does it incorrectly [2], thus opening an avenue for a host of attacks exploiting for vulnerabilities.

Successful XSS can result in serious security violations for both the web site and the user. An attacker can inject a malicious code into where a web application accepts user input, and if the input is not validated, the code can steal cookies, transfer private information, hijack a user's account, manipulate the web content, cause denial of service, and many other malicious activities [2,3].

XSS attacks are of three types namely reflected, stored and DOM-based [2,3]. Reflected XSS is executed by the victim's browser and occurs when the victim provides input to the web site. Stored XSS attacks store the malicious script in databases, message forums, comments fields, etc. of the attacked server. The malicious script is executed by visiting users thereby passing their privileges to the attacker. Both reflected and stored XSS are executed on the server side. On the other hand, DOM-based XSS attacks are executed on the client side. Attackers are able to collect sensitive or important information from the user's computer.

The purpose of this paper is to show the results of the systematic literature review we conducted on the current state of research on XSS. The review covers the period since XSS was first discovered up to the end of 2012. The rest of the paper is structured as follows. Section 2 describes the method we used to conduct the study and we present the results in Section 3. We answer our research questions in Section 4 and conclude the paper in Section 5.

2. Research method

This study is a systematic literature review of research studies on XSS. It was performed following the guidelines provided by Kitchenham [4]. We used the Mendeley reference manager [5] for storing and organizing the studies, and for referencing.

2.1. Research questions

The research questions we addressed in this study are as below:

RQ1: How much research has been done on XSS since its discovery?
RQ2: What are the proposed techniques or solutions to address the issue of XSS?
RQ3: On which area(s) is research on XSS mostly focused?
RQ4: Which of the three types of XSS is addressed the most?

To answer RQ1, we decided to look at all publications from 2000 since that was the year XSS vulnerabilities were first announced to the public. However, from our findings academic research on XSS pick up speed from 2004 onwards.

From this study, we also wanted to know what are the proposed techniques or solutions (RQ2) that exist so far to address the problem of XSS. We identified the techniques, tools, methods, and algorithms that each article provided.

With respect to RQ3, we wanted to know what type of solution is being proposed, whether it prevents XSS attacks or vulnerabilities or detect them when they occur in a program, or better still remove the vulnerabilities from the program.

To address RQ4, we looked at the type of XSS vulnerabilities that each article addressed to determine if they included Reflected XSS, Stored XSS, or DOM-based XSS, or all three.

2.2. Search process

A search of online databases was carried out to collect articles for this study. These databases are known to contain published work in our field of interest and have been used by many researchers conducting systematic literature reviews in software engineering. Many articles were downloaded from each database based on their relevance to our search terms. The databases and their URLs are shown in Table 1. The search terms we used included:

- Cross site scripting.
- Cross-site scripting.
- Cross site scripting attack.
- Cross site scripting vulnerability.
- XSS.
- XSS attacks.
- XSS vulnerabilities.
- Software security vulnerabilities.
- Web application vulnerabilities.
- Web application security problems.

Table 1
Online sources searched for relevant studies.

Database	URL
IEEE Explore	http://ieeexplore.ieee.org/Xplore/home.jsp
ScienceDirect	www.sciencedirect.com
ACM Digital Library	www.acm.org/dl
SpringerLink	www.springerlink.com
Google Scholar	http://scholar.google.com/
CiteseerX	http://citeseerx.ist.psu.edu/index

Please cite this article in press as: I. Hydara et al., Current state of research on cross-site scripting (XSS) – A systematic literature review, Inform. Softw. Technol. (2014), http://dx.doi.org/10.1016/j.infsof.2014.07.010

Table 2
Quality score of selected studies.

Study	QA1	QA2	QA3	Total score
S001	P	Y	Y	2.5
S002	P	Y	Y	2.5
S003	P	P	P	1.5
S004	P	P	Y	2
S005	Y	P	Y	2.5
S006	Y	Y	Y	3
S007	Y	Y	Y	3
S008	Y	Y	Y	3
S009	Y	P	Y	2.5
S010	Y	Y	Y	3
S011	Y	Y	Y	3
S012	Y	Y	Y	3
S013	P	Y	Y	2.5
S014	Y	Y	Y	3
S015	P	Y	Y	2.5
S016	P	Y	Y	2.5
S017	Y	P	Y	2.5
S018	P	Y	Y	2.5
S019	P	Y	Y	2.5
S020	P	P	P	1.5
S021	P	P	Y	2
S022	Y	Y	Y	3
S023	Y	Y	Y	3
S024	Y	Y	Y	3
S025	P	Y	Y	2.5
S026	P	Y	Y	2.5
S027	Y	Y	Y	3
S028	Y	Y	Y	3
S029	P	Y	Y	2.5
S030	Y	Y	Y	3
S031	Y	Y	Y	3
S032	Y	Y	Y	3
S033	P	Y	Y	2.5
S034	P	Y	Y	2.5
S035	Y	Y	Y	3
S036	Y	Y	Y	3
S037	Y	Y	Y	3
S038	Y	Y	Y	3
S039	Y	Y	Y	3
S040	P	Y	Y	2.5
S041	Y	P	Y	2.5
S042	Y	Y	Y	3
S043	Y	Y	Y	3
S044	P	Y	Y	2.5
S045	P	Y	Y	2.5
S046	P	Y	Y	2.5
S047	P	Y	Y	2.5
S048	Y	Y	Y	3
S049	Y	Y	Y	3
S050	P	Y	Y	2.5
S051	Y	Y	Y	3
S052	P	Y	Y	2.5
S053	Y	Y	Y	3
S054	P	Y	Y	2.5
S055	Y	Y	Y	3
S056	P	Y	Y	2.5
S057	Y	Y	Y	3
S058	Y	P	Y	2.5
S059	P	Y	Y	2.5
S060	P	Y	Y	2.5
S061	Y	Y	Y	3
S062	P	Y	Y	2.5
S063	P	P	Y	2
S064	Y	Y	Y	3
S065	Y	P	Y	2.5
S066	Y	Y	Y	3
S067	P	Y	Y	2.5
S068	P	P	Y	2
S069	Y	P	Y	2.5
S070	P	P	Y	2
S071	Y	P	Y	2.5
S072	Y	Y	Y	3
S073	P	Y	Y	2.5
S074	Y	Y	Y	3
S075	Y	Y	Y	3
S076	Y	Y	Y	3
S077	Y	Y	Y	3
S078	Y	Y	Y	3
S079	Y	Y	Y	3
S080	Y	Y	Y	3
S081	Y	Y	Y	3
S082	Y	Y	Y	3
S083	Y	Y	Y	3
S084	P	Y	Y	2.5
S085	P	Y	Y	2.5
S086	P	Y	Y	2.5
S087	Y	Y	Y	3
S088	P	Y	Y	2.5
S089	P	Y	Y	2.5
S090	Y	Y	Y	3
S091	Y	Y	Y	3
S092	Y	Y	Y	3
S093	Y	Y	Y	3
S094	Y	Y	Y	3
S095	Y	Y	Y	3
S096	Y	Y	Y	3
S097	Y	Y	Y	3
S098	P	Y	Y	2.5
S099	Y	Y	Y	3
S100	Y	Y	Y	3
S101	Y	Y	Y	3
S102	P	Y	Y	2.5
S103	Y	Y	Y	3
S104	Y	Y	Y	3
S105	Y	Y	Y	3
S106	Y	Y	Y	3
S107	Y	Y	Y	3
S108	Y	Y	Y	3
S109	P	Y	Y	2.5
S110	P	Y	Y	2.5
S111	P	Y	Y	2.5
S112	P	Y	Y	2.5
S113	P	Y	Y	2.5
S114	Y	Y	Y	3
S115	Y	Y	Y	3

We also combined some of the search terms using Boolean AND/OR.

In addition, we looked at the references of some of the downloaded articles and searched for the referenced publications that have titles related to our topic of interest. This was done in the hope of obtaining more publications that were not available in the online databases we searched.

2.3. Inclusion and exclusion criteria

Articles that met the following criteria were included:

- Peer-reviewed articles that focused on the problem of XSS and published before January 2013.
- Articles that address XSS alongside other security vulnerabilities such as SQL injection.
- Articles that described proposed tools to address the problem of XSS.

Article on the following topics were excluded:

- Survey papers on XSS.
- Articles where XSS is only discussed as an example of security vulnerability and is not the focus of the research.
- White papers on XSS.
- Book chapters on XSS.

2.4. Quality assessment

The next step after using the inclusion and exclusion criteria was to conduct the quality assessment of the remaining papers. Each paper was evaluated following the York University, Centre for Reviews of Dissemination (CRD) Database of Abstracts of Reviews of Effects (DARE) criteria as explained by Kitchenham [4]. The following questions were set to assess the quality of the papers for this study:

QA1: Is the research focused on XSS vulnerabilities?
QA2: Are the research problem(s) clearly stated?
QA3: Is the proposed solution clearly explained?

The quality questions were scored as follows:

QA1: Y (yes), the research focused on XSS vulnerabilities; P (partly), the research addressed XSS and another related vulnerability; N (no), the research did not focus on XSS.
QA2: Y, the research problem(s) are clearly stated; P, research problem(s) were stated but not clearly explained; N, research problem(s) were not stated.
QA3: Y, the proposed solution was explained clearly; P, the proposed solution was briefly described; N, the proposed solution was not clearly explained.

We used the following procedure to score the quality assessment of each paper: $Y = 1$; $P = 0.5$; $N = 0$. Table 2 shows the quality scores of the studies selected for this review.

2.5. Data collection

The data collected from each selected paper were as follows:

- The author(s).
- The title of the paper.
- The year of publication.
- The journal/ proceeding in which it was published and full reference.
- The problem statement/aim of study.
- The proposed solution and its details.
- The type of solution (prevention, detection, or removal).
- The type of XSS addressed (reflected, stored, or DOM-based).

2.6. Data analysis

The data was tabulated as follows:

- The number of research papers published per year and their source (RQ1).
- The proposed solution of each paper (RQ2).
- Whether the solution is to prevent, detect, or remove XSS vulnerabilities (RQ3).
- The categories of XSS vulnerabilities addressed, whether reflected, stored, or DOM-based (RQ4).

3. Results

In this section, we summarize the results of our study.

In Table 2, we show the results of the quality assessment of the studies. The studies were reviewed based on the quality assessment questions and given a score for each question. The last column indicates the total score for each study.

Table 3 shows the 115 studies we selected from the review with the data we need to answer our first research question. It details the author(s) of each study, the title of the study, the year it was published in, and the source of publication, be it a journal or conference proceeding. The studies are arranged alphabetically based on the authors' names and each study was labelled uniquely from S001 to S115. Fourteen of the studies have been published in journals and 101 studies have been published in conference proceedings. Fig. 1 shows the distribution of studies based on the year of publications. Table 4 gives the names of the journals and conferences in which the studies were published and how many studies were published by each journal and conference.

Table 5 shows the summary of our findings from the studies that will help answer our research questions RQ2–RQ4. It details the proposed techniques or solution for each study in the second column. This is taken verbatim form the studies' authors. It also shows the area of focus for each study as well as the type of XSS in the third and fourth columns, respectively. The area of focus tells us whether each study's solution is geared towards XSS attack prevention, detection or implementation, or XSS vulnerability detection, prediction, or removal, or a combination of any of these. As for the XSS categories addressed, we identified them as reflected, stored, or DOM-based as specified by the authors. Where a study did not indicate the type of XSS, we filled it with “Not specified” and “All” indicates that a study addresses all the three categories.

Table 6 summarises the techniques/solutions proposed by the studies. The techniques are categorised under techniques/solutions in the first column and the second column indicates the studies that fall under each category. The third column shows the percentage of the studies in each category.

Table 7 shows a comparison between dates and proposed techniques/solutions of the studies.

To determine the area of focus of the studies, we identified six groups into which the studies were categorized. The groups are stated in the first column of Table 8 with the number of studies falling in each group in the second column. The last column indicates the percentage of studies in each group.

Similar to Table 8, we identified the number of studies that addressed each of the types of XSS, namely reflected, stored and DOM-based XSS in Table 9. We also identified those studies that addressed more than one or all the categories. For those studies where the author(s) did not indicate the categories of XSS addressed we identified them as not specified.

4. Discussion

In this section, we discuss the answers to our research questions.

4.1. How much research has been done on XSS since 2000?

In the overall study, we identified 115 relevant studies from the sources we searched as shown in Table 2. The data shows that since 2004 research on XSS has been growing steadily as indicated by Fig. 1. The number of papers published is seen to have been increasing yearly, except for 2008 and 2012. This shows that research on XSS is very active and still going on.

Also, publications have been diverse across many journals and conference proceedings as we can see in Table 4. Thirteen of the studies are published in 11 different journals, most of which are of very high reputations belonging to ACM, IEEE and Elsevier. They cover different areas of Computer Science such as Software Engineering, Security, Networking, etc. The other 101 studies are published in 72 different conference proceedings, including ACM and IEEE conferences, as well as many other international conferences of high repute. This demonstrates the importance of security research in Software Engineering and other related fields.

Table 3
The systematic review studies and their sources.

Study	Author(s)	Title	Year	Journal/Proceeding
S001	Adi [6]	A design of a proxy inspired from human immune system to detect SQL injection and cross-site scripting	2012	Procedia Engineering
S002	Agosta et al. [7]	Automated security analysis of dynamic web applications through symbolic code execution	2012	2012 Ninth International Conference on Information Technology – New Generations
S003	Al-Amro and El-Qawasmeh [8]	Discovering security vulnerabilities and leaks in ASP. NET websites	2012	Cyber Security, Cyber Warfare and Digital Forensic (CyberSec), 2012 International Conference on
S004	Arulsuju [9]	Hunting malicious attacks in social networks	2011	Advanced Computing (ICoAC), 2011 Third International Conference on
S005	Athanasopoulos et al. [10]	Hunting cross-site scripting attacks in the network	2010	W2SP 2010: Web 2.0 Security and Privacy Workshop 2010
S006	Avancini and Ceccato [11]	Security testing of web applications: a search-based approach for cross-site scripting vulnerabilities	2011	2011 IEEE 11th International Working Conference on Source Code Analysis and Manipulation
S007	Avancini and Ceccato [12]	Towards security testing with taint analysis and genetic algorithms	2010	Proceedings of the 2010 ICSE Workshop on Software Engineering for Secure Systems
S008	Avancini and Ceccato [13]	Grammar based Oracle for security testing of web applications	2012	2012 7th International Workshop on Automation of Software Test (AST)
S009	Barhoom and Kohail [14]	A new server-side solution for detecting cross site scripting attack	2011	International journal of Computer Information Systems
S010	Barth [15]	Secure content sniffing for web browsers, or how to stop papers from reviewing themselves	2009	2009 30th IEEE Symposium on Security and Privacy
S011	Bates et al. [16]	Regular expressions considered harmful in client-side XSS filters	2010	Proceedings of the 19th International Conference on World Wide Web, WWW'10
S012	Bathia et al. [17]	Assisting programmers resolving vulnerabilities in Java web applications	2011	CCIS 2011: Communications in Computer and Information Science
S013	Bencsath et al. [18]	XCS based hidden firmware modification on embedded devices	2011	SoftCOM 2011 19th International Conference on Software Telecommunications and Computer Networks
S014	Bisht and Venkatakrishnan [19]	XSS-GUARD Precise dynamic prevention of cross-site scripting attacks	2008	Lecture Notes in Computer Science
S015	Bojinov et al. [20]	XCS: cross channel scripting and its impact on web applications	2009	CCS '09: Proceedings of the 16th ACM conference on Computer and communications security
S016	Brinhosa et al. [21]	Proposal and development of the web services input validation model	2012	2012 IEEE Network Operations and Management Symposium (NOMS)
S017	Cao et al. [22]	POSTER: A path-cutting approach to blocking XSS worms in social web networks	2011	CCS '11: Proceedings of the 18th ACM conference on Computer and communications security
S018	Chaudhuri and Foster [23]	Symbolic security analysis of ruby-on-rails web applications	2010	Proceedings of the 17th ACM conference on Computer and communications security - CCS '10
S019	Chen and Wu [24]	An automated vulnerability scanner for injection attack based on injection point	2010	2010 International Computer Symposium ICS2010
S020	Choi et al. [25]	Efficient malicious code detection using N-Gram analysis and SVM	2011	2011 14th International Conference on Network Based Information Systems
S021	Coppolino et al. [26]	From Intrusion detection to intrusion detection and diagnosis: an ontology-based approach	2009	Lecture Notes in Computer Science
S022	Di Licca et al. [27]	Identifying cross site scripting vulnerabilities in web applications	2004	26th Annual International Telecommunications Energy Conference
S023	Duchene et al. [28]	XSS vulnerability detection using model inference assisted evolutionary fuzzing	2012	2012 IEEE Fifth International Conference on Software Testing, Verification and Validation
S024	Faghani and Saidi [29]	Social networks' XSS worms	2009	2009 International Conference on Computational Science and Engineering
S025	Fonseca et al. [30]	Testing and comparing web vulnerability scanning tools for SQL injection and XSS attacks	2007	13th Pacific Rim International Symposium on Dependable Computing (PRDC 2007)
S026	Frenz and Yoon [31]	XSSmon: A Perl based IDS for the detection of potential XSS attacks	2012	2012 IEEE Long Island Systems, Application and Technology Conference (LISAT)
S027	Galan et al. [32]	A multi-agent scanner to detect stored-XSS vulnerabilities	2010	2010 International Conference for Internet Technology and Secured Transactions (ICITST)
S028	Garcia-Alfaro and Navarro-Arribas [33]	Prevention of cross-site scripting attacks on current web applications	2007	OTM'07: Proceedings of the 2007 OTM confederated international conference on the move to meaningful internet systems
S029	Gilad and Herzberg [34]	Off-path attacking the web	2012	Proceedings of the 6th USENIX conference on Offensive Technologies
S030	Grabowski et al. [35]	Type-based enforcement of secure programming guidelines – code injection prevention at SAP	2012	Lecture Notes in Computer Science
S031	Gundy and Chen [36]	Noncespaces: Using randomization to enforce information flow tracking and thwart cross-site scripting attacks	2009	16th Annual Network and Distributed System Security Symposium Proceedings, NDSS 2009
S032	Heiderich and Holz [37]	Crouching Tiger – hidden payload: security risks of scalable vectors graphics	2011	CCS '11: Proceedings of the 18th ACM conference on Computer and communications security
S033	Hermosillo et al. [38]	AProSec: an aspect for programming secure web applications	2007	Second International Conference on Availability, Reliability and Security (ARES'07)
S034	Hidhaya and Geetha [39]	Intrusion protection against SQL injection and cross site scripting attacks using a reverse proxy	2012	Communications in Computer and Information Science
S035	Hooimeijer et al. [40]	Fast and precise sanitizer analysis with BEK	2011	SEC'11: Proceedings of the 20th USENIX conference on Security
S036	Iha and Doi [41]	An implementation of the binding mechanism in the web browser for preventing XSS Attacks: introducing the bind-value	2009	2009 International Conference on Availability Reliability and Security

(continued on next page)

Table 3 (*continued*)

Study	Author(s)	Title	Year	Journal/Proceeding
		headers		
S037	Ismail et al. [42]	A proposal and implementation of automatic detection/ collection system for cross-site scripting vulnerability	2004	18th International Conference on Advanced Information Networking and Applications, AINA 2004
S038	Jayamsakthi and Ponnavaikko [43]	Risk mitigation for cross site scripting attacks using signature based model on the server side	2007	Second International Multi-Symposiums on Computer and Computational Sciences (IMSCCS 2007)
S039	Johns [44]	SessionSafe: Implementing XSS immune session handling	2006	Lecture Notes in Computer Science
S040	Johns et al. [45]	Secure Code Generation for Web Applications	2010	Lecture Notes in Computer Science
S041	Johns et al. [46]	XSSDS: Server-side Detection of Cross-site Scripting Attacks	2008	2008 Annual Computer Security Applications Conference
S042	Jovanovic et al. [47]	Pixy: A Static Analysis Tool for Detecting Web Application Vulnerabilities (Short Paper)	2006	Proceedings of the 2006 IEEE Symposium on Security and Privacy
S043	Jovanovic et al. [48]	Precise Alias Analysis for Static Detection of Web Application Vulnerabilities	2006	PLAS ‘06: Proceedings of the 2006 workshop on Programming languages and analysis for security
S044	Juillerat [49]	Enforcing code security in database web applications using libraries and object models	2007	Proceedings of the 2007 Symposium on Library-Centric Software Design - LCSD ‘07
S045	Kals et al. [50]	SecuBat: A Web Vulnerability Scanner	2006	WWW ‘06: Proceedings of the 15th international conference on World Wide Web
S046	Kerschbaum [51]	Simple cross-site attack prevention	2007	2007 Third International Conference on Security and Privacy in Communications Networks and the Workshops - SecureComm 2007
S047	Kieyzun et al. [52]	Automatic creation of SQL Injection and cross-site scripting attacks	2009	2009 IEEE 31st International Conference on Software Engineering
S048	Kirda et al. [53]	Client-side cross-site scripting protection	2009	Computers & Security
S049	Kirda et al. [54]	Noxes: A Client-side solution for mitigating cross-site scripting attacks	2006	SAC ‘06: Proceedings of the 2006 ACM symposium on Applied computing
S050	Komiya et al. [55]	Classification of malicious web code by machine learning	2011	011 3rd International Conference on Awareness Science and Technology iCAST
S051	Li [56]	Towards security vulnerability detection by source code model checking	2010	Software Testing Verification and Validation Workshops ICSTW 2010 Third International Conference on
S052	Li et al. [57]	Perturbation-based user-input-validation testing of web applications	2010	Journal of Systems and Software
S053	Li and Wang [58]	FIRM: Capability-based Inline Mediation of Flash Behaviors	2010	ACSAC ‘10: Proceedings of the 26th Annual Computer Security Applications Conference
S054	Livshits and Erlingsson [59]	Using web application construction frameworks to protect against code injection attacks	2007	Proceedings of the 2007 workshop on Programming languages and analysis for security - PLAS ‘07
S055	Louw and Venkatakrishnan [60]	Blueprint: robust prevention of cross-site scripting attacks for existing browsers	2009	2009 30th IEEE Symposium on Security and Privacy
S056	Martin and Lam [61]	Automatic generation of XSS and SQL injection attacks with goal-directed model checking	2008	17th Conference on Security Symposium, 2008
S057	McAllister et al. [62]	Leveraging user interactions for in-depth testing of web applications	2008	Lecture Notes in Computer Science
S058	Minamide [63]	Static approximation of dynamically generated web pages	2005	WWW ‘05: Proceedings of the 14th international conference on World Wide Web
S059	Mohosina and Zulkernine [64]	DESERVE: A framework for detecting program security vulnerability exploitations	2012	2012 IEEE Sixth International Conference on Software Security and Reliability
S060	Mui and Frankl [65]	Preventing web application injections with complementary character coding	2011	Lecture Notes in Computer Science
S061	Nadji et al. [66]	Document structure integrity: a robust basis for cross-site scripting defence	2009	16th Annual Network and Distributed System Security Symposium, NDSS 2009
S062	Nanda et al. [67]	Dynamic multi-process information flow tracking for web application security	2007	Proceedings of the 8th ACM/IFIP/USENIX international conference on Middleware - Middleware ‘07
S063	Nguyen-Tuong [68]	Automatically hardening web applications using precise tainting	2005	IFIP Advances in Communication and Information Technology
S064	Nikiforakis et al. [69]	SessionShield: Lightweight protection against session Hijacking	2011	ESSoS'11: Proceedings of the Third international conference on Engineering secure software and systems
S065	Nunan et al. [70]	Automatic classification of cross-site scripting in web pages using document-based and URL-based features	2012	Computers and Communications (ISCC), 2012 IEEE Symposium on
S066	Pelizzi and Sekar [71]	Protection, usability and improvements in reflected XSS filters	2012	Proceedings of the 7th ACM Symposium on Information, Computer and Communications Security, ASIACCS'12
S067	Perez et al. [72]	LAPSE + Static analysis security software: vulnerabilities detection in Java EE applications	2011	Communications in Computer and Information Science
S068	Petkov [73]	Overcoming programming flaws: indexing of common software vulnerabilities	2005	**InfoSecCD ‘05**:Proceedings of the 2nd annual conference on Information security curriculum development
S069	Phung et al. [74]	Lightweight self-protecting JavaScript	2009	ASIACCS ‘09: Proceedings of the 4th International Symposium on Information, Computer, and Communications Security
S070	Priyadarshini et al. [75]	A cross platform intrusion detection system using inter server communication technique	2011	2011 International Conference on Recent Trends in Information Technology ICRTIT
S071	Putthacharoen and Bunyatnoparat [76]	Protecting cookies from cross site script attacks using dynamic cookies rewriting technique	2011	13th International Conference on Advanced Communication Technology ICACT2011
S072	Saxena et al. [77]	ScriptGard: Automatic context-sensitive sanitization for large-scale legacy web applications categories and subject descriptors	2011	CCS ‘11: Proceedings of the 18th ACM conference on Computer and communications security
S073	Scholte et al. [78]	Preventing input validation vulnerabilities in web applications through automated type analysis	2012	2012 IEEE 36th Annual Computer Software and Applications Conference
S074	Shahriar and	Injecting comments to detect JavaScript code injection attacks	2011	2011 IEEE 35th Annual Computer Software and Applications

Table 3 (*continued*)

Study	Author(s)	Title	Year	Journal/Proceeding
	Zulkernine [79]			Conference Workshops
S075	Shahriar and Zulkernine [80]	MUTEC: Mutation-based Testing of Cross Site Scripting School of Computing	2009	Software Engineering for Secure Systems, 2009. SESS ‘09. ICSE Workshop on
S076	Shahriar and Zulkernine [81]	S2XS2: A Server Side Approach to Automatically Detect XSS Attacks	2011	2011 IEEE Ninth International Conference on Dependable, Autonomic and Secure Computing
S077	Shahriar and Zulkernine [82]	Trustworthiness testing of phishing websites: A behavior model-based approach	2012	Future Generation Computer Systems
S078	Shanmugam and Ponnavaikko [83]	A solution to block cross site scripting vulnerabilities based on service oriented architecture	2007	6th IEEE/ACIS International Conference on Computer and Information Science (ICIS 2007)
S079	Shanmugam and Ponnavaikko [84]	Behavior-based anomaly detection on the server side to reduce the effectiveness of cross site scripting vulnerabilities	2007	Third International Conference on Semantics, Knowledge and Grid (SKG 2007)
S080	Shanmugam and Ponnavaikko [85]	XSS application worms: new internet infestation and optimized protective measures	2007	Eighth ACIS International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/ Distributed Computing (SNPD 2007)
S081	Shar and Tan [86]	Auditing the defense against cross site scripting in web applications	2010	2010 International Conference on Security and Cryptography SECRYPT
S082	Shar and Tan [87]	Auditing the XSS defence features implemented in web application programs	2012	IET Software
S083	Shar and Tan [88]	Automated removal of cross site scripting vulnerabilities in web applications	2012	Information and Software Technology
S084	Shar and Tan [89]	Mining input sanitization patterns for predicting SQL injection and cross site scripting vulnerabilities	2012	Proceedings - 34th International Conference on Software Engineering, ICSE 2012
S085	Shar and Tan [90]	Predicting common web application vulnerabilities from input validation and sanitization code patterns	2012	Proceedings of the 27th IEEE/ACM International Conference on Automated Software Engineering - ASE 2012
S086	Sharma et al. [91]	Integrated approach to prevent SQL injection attack and reflected cross site scripting attack	2012	International Journal of System Assurance Engineering and Management
S087	Sivakumar and Garg [92]	Constructing a “Common Cross Site Scripting Vulnerabilities Enumeration (CXE)” Using CWE and CVE	2007	Lecture Notes in Computer Science
S088	Somorovsky et al. [93]	All Your Clouds are Belong to us – Security Analysis of Cloud Management Interfaces	2011	CCSW ‘11: Proceedings of the 3rd ACM workshop on Cloud computing security workshop
S089	Stuckman and Purtilo [94]	A Testbed for the Evaluation of Web Intrusion Prevention Systems	2011	011 Third International Workshop on Security Measurements and Metrics
S090	Sun et al. [95]	Client-Side detection of XSS worms by monitoring payload propoagation	2009	Lecture Notes in Computer Science
S091	Sun and He [96]	Model checking for the defense against cross-site scripting attacks	2012	2012 International Conference on Computer Science and Service System
S092	Sundareswaran and Squicciarini [97]	XSS-Dec: a hybrid solution to mitigate cross-site scripting attacks	2012	Lecture Notes in Computer Science
S093	Takesue [98]	A Protection Scheme against the Attacks Deployed by Hiding the Violation of the Same Origin Policy	2008	2008 Second International Conference on Emerging Security Information Systems and Technologies
S094	Tang et al. [99]	Alhambra: a system for creating, enforcing, and testing browser security policies	2010	WWW ‘10: Proceedings of the 19th international conference on World wide web
S095	Tang et al. [100]	L-WMxD: lexical based webmail XSS discoverer	2011	2011 IEEE Conference on Computer Communications Workshops INFOCOM WKSHPS
S096	Shalini and Usha [101]	Prevention of cross-site scripting attacks (XSS) on web applications in the client side	2011	International Journal of Computer Science Issues
S097	Tiwari et al. [102]	Optimized client side solution for cross site scripting	2008	2008 16th IEEE International Conference on Networks
S098	Tsai et al. [103]	Optimum tuning of defense settings for common attacks on the web applications	2009	3rd Annual 2009 International Carnahan Conference on Security Technology
S099	V. Sharath Chandra and Selvakumar [104]	Bixsan: Browser Independent XSS Sanitizer for prevention of XSS attacks	2011	ACM SIGSOFT Software Engineering Notes
S100	Van Gundy and Chen [105]	Noncespaces: Using randomization to defeat cross-site scripting attacks	2012	Computers & Security
S101	Van-Acker et al. [106]	FlashOver: Automated Discovery of Cross-site Scripting Vulnerabilities in Rich Internet Applications	2012	ASIACCS ‘12: Proceedings of the 7th ACM Symposium on Information, Computer and Communications Security
S102	Venkatakrishnan et al. [107]	WebAppArmor: a framework for robust prevention of attacks on Web4Applications (Invited Paper)	2010	Lecture Notes in Computer Science
S103	Vogt et al. [108]	Cross-site scripting prevention with Dyna6ic data tainting and static analysis	2007	NDSS’07: Network and Distributed System Security Symposium
S104	Wang et al. [109]	Investigations in cross-site script on web-systems gathering digital evidence against cyber-intrusions	2007	Future Generation Communication and Networking (FGCN 2007) (Volume:2)
S105	Wang et al. [110]	Program slicing stored XSS bugs in web application	2011	2011 Fifth International Conference on Theoretical Aspects of Software Engineering
S106	Wassermann and Su [111]	Static Detection of Cross-Site Scripting Vulnerabilities	2008	ICSE ‘08: Proceedings of the 30th international conference on Software engineering
S107	Weinberger et al. [112]	A Systematic Analysis of XSS Sanitization in Web Application Frameworks	2011	ESORICS’11: Proceedings of the 16th European conference on Research in computer security
S108	Wurzinger et al. [113]	SWAP: mitigating XSS attacks using a reverse proxy	2009	Software Engineering for Secure Systems, 2009. SESS ‘09. ICSE Workshop on
S109	Xin-hua and Zhi-jian [114]	A static analysis tool for detecting web application injection vulnerabilities for ASP program	2010	2nd International Conference on e-Business and Information Security (EBISS)
S110	Xiong et al. [115]	Model-based penetration test framework for web applications using TTCN-3	2009	Lecture Notes in Business Information Processing
S111	Yu et al. [116]	STRANGER: an automata-based string analysis tool for PHP	2010	Lecture Notes in Computer Science

(*continued on next page*)

Table 3 (*continued*)

Study	Author(s)	Title	Year	Journal/Proceeding
S112	Yu et al. [117]	String abstractions for string verification	2011	Lecture Notes in Computer Science
S113	Zhang et al. [118]	D-WAV: a web application vulnerabilities detection tool using characteristics of web forms	2010	Software Engineering Advances ICSEA 2010 Fifth International Conference on
S114	Zhang et al. [119]	An execution-flow based method for detecting Cross-site Scripting attacks	2010	Software Engineering and Data Mining SEDM 2010 2nd International Conference on
S115	Zhenyu et al. [120]	MBDS: model-based detection system for cross site scripting	2007	IET Conference on Wireless, Mobile and Sensor Networks, 2007

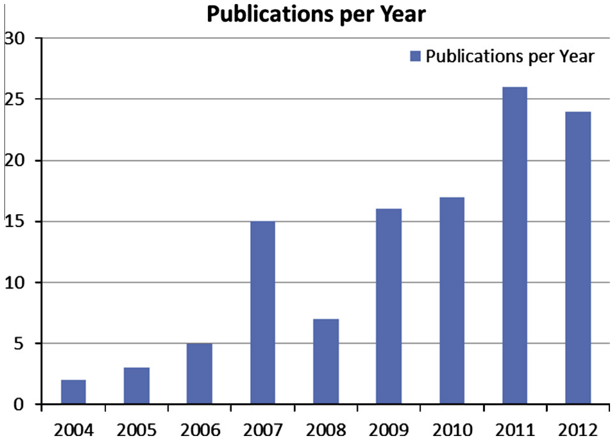

Fig. 1. The number of publications per year.

We found that 87.8% of the studies were published in conference proceedings. This may be attributed to the fact that computing research publications in conferences usually have faster turnaround times than in journals [121]. Conferences enable faster dissemination of knowledge to the intended audiences, which is very important in security research [121]. Another factor may be the possibility of meeting and sharing directly with other researchers and getting immediate feedback to improve on results.

References [122,123] believe that computer science research should follow older disciplines and publish broader and more detailed papers in journals instead of the short, fast papers in conferences. To address the issue of slow publication, the authors suggest the computer science community should adopt the usage of online archives like in other fields. This enables fast dissemination of the information to be published and allows research time to prepare detailed papers for journal publication.

However, it is worth noting though that journals publish more thorough and detailed research than most conferences do. Hence it takes more time to prepare a paper for a journal publication. In most conferences, selected papers' authors are requested to submit extended versions of their papers to be published in a journal. In addition, research on web application security, in general, and on XSS, in particular, is very recent. Therefore, it will take time before a lot of publications on XSS are found in journals.

4.2. What are the proposed techniques to address the issue of XSS?

The proposed techniques/solutions suggested by the studies are many and varied. They range from static and dynamic analysis, to modelling, secure programming, etc. We will discuss them under the following headings.

4.2.1. Static analysis

it involves reviewing the source code or byte code of an application in order to find faults [72]. As seen in Table 6, twenty-seven studies (23.5%) proposed static analysis techniques as solutions to XSS problems. Static taint analysis, a technique which tracts tainted values through the control flow graph [11], was proposed in [7,11,12,77,83,111,119]. Most of the studies used more than one technique in their proposed solutions. In [7], static taint analysis is combined with symbolic code execution, in [88], in [11,12] it is combined with genetic algorithms, and it is combined with string analysis in [111]. Other techniques include program slicing [17,64,110], symbolic execution [23], data flow analysis [47], string analysis [116], and precise alias analysis [48].

4.2.2. Dynamic analysis

On the other hand dynamic analysis entails examining the behaviour of an application in runtime [72]. It is proposed in 57 of the studies (49.6%) as shown in Table 6. The dynamic analysis techniques proposed comprised of black-box testing [24,50,62,93], taint tracking [52,65–68,99], flow analysis [97,119], monitoring [46,61], filtering [19], and dynamic analysis [118]. Five studies [27,74,106–108] combined both static analysis and dynamic analysis techniques.

4.2.3. Secure programming

Secure programming techniques are proposed in 3 of the studies (2.6%). These techniques ensure that programming guidelines and rules are followed during the development of an application. In [35] a technique called Type Systems is used to automatically enforce programming guidelines, while [45] used ELET (Embedded Language Encapsulation Type) to enforce secure code generation in programming languages. Libraries and Object Models are used in [49] to also enforce secure coding in database web applications.

4.2.4. Modelling

Models were proposed by another 18 studies (15.7%). The models proposed were based on the following techniques and approaches: abstraction [13,117], model checking [56,96], model inference and evolutionary fuzzing [28], input validation [21,86,90], simulation [29], signature based model [43], deferred loading, one-time URLs, and subdomain switching [44], threading [85], control flow graph [87], data mining [89], hybrid approach [91], TTCN-3 [115], Finite State Machine [82], and primitive and advanced models [120].

4.2.5. Others

The remaining 5 studies whose proposed techniques did not fall into the categories discussed above were put under the 'Others' category. They focus on benchmarking [30,92,94,109] and indexing [73] of XSS vulnerabilities for facilitating further research on XSS issues.

There is no single solution, so far, that can eliminate XSS vulnerabilities and prevent XSS attacks. Therefore, a number of mitigation techniques should be employed to curb the spread of the XSS attacks and eliminate XSS vulnerabilities. Dynamic analysis still remains the leading approach to tackle XSS vulnerabilities and attacks as evidenced by the academic studies we reviewed (see Table 7) and suggested from the industry side [124–129].

Table 4
Names of journals and conferences and the number of studies published in each.

No.	Title	No. of studies
Journals		
1	ACM SIGSOFT Software Engineering Notes	1
2	Computers and Security	2
3	Communications in Computer and Information Science	2
4	Future Generation Communication and Networking	1
5	Future Generation Computer Systems	1
6	IET Software	1
7	Information and Software Technology	1
8	International Journal of Computer Information System	1
9	International Journal of Computer Science Issues	1
10	International Journal of System Assurance Engineering and Management	1
11	Journal of Systems and Software	1
12	Procedia Engineering	1
Conference proceedings		
1	ACIS International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing	1
2	ACM Conference on Computer and Communications Security	5
3	ACM Symposium on Applied Computing	1
4	ACM Symposium on Information, Computer and Communications Security	3
5	ACM Workshop on Cloud Computing Security	1
6	ACM/IFIP/USENIX International Conference on Middleware	1
7	Communications in Computer and Information Science	1
8	Computer Security Applications Conference	2
9	Conference on Information Security Curriculum Development	1
10	DBSec	1
11	DIMVA	1
12	European Conference on Research in Computer Security	4
13	FAST	1
14	ICISS	2
15	IEEE Computer Software and Applications Conference	2
16	IEEE Conference on Computer Communications	1
17	IEEE International Conference on Dependable, Autonomic and Secure Computing	1
18	IEEE International Conference on Networks	1
19	IEEE International Conference on Recent Trends in Information Technology	1
20	IEEE International Conference on Software Security and Reliability	1
21	IEEE International Conference on Software Testing, Verification and Validation	2
22	IEEE Long Island Systems, Application and Technology Conference	1
23	IEEE Network Operations and management Symposium	1
24	IEEE Symposium on Computers and Communications	1
25	IEEE Symposium on Security and Privacy	3
26	IEEE/ACIS international Conference on Computer and Information Science	1
27	IEEE/ACM International Conference on Automated Software Engineering	1
28	IFIP Advances in Communication and Information Technology	1
29	International Carnahan Conference on Security Technology	1
30	International Computer Symposium	1
31	International Conference for Internet Technology and Secured Transactions	1
32	International Conference on Advanced Communication Technology	1
33	International Conference on Advanced Computing	1
34	International Conference on Advanced Information Networking and Applications	1
35	International Conference on Availability, Reliability and Security	2
36	International Conference on Awareness Science and Technology	1
37	International Conference on Computational Science and Engineering	1
38	International Conference on Computer Science and Service System	1
39	International Conference on Cyber Security, Cyber Warfare and Digital Forensic	1
40	International Conference on e-Business and Information Security	1
41	International Conference on Emerging Security Information Systems and Technologies	1
42	International Conference on Engineering Secure Software and Systems	2
43	International Conference on Information Technology	1
44	International Conference on Network Based Information Systems	1
45	International Conference on Security and Cryptography	1
46	International Conference on Security and Privacy	1
47	International Conference on Semantics, Knowledge and Grid	1
48	International Conference on Software Engineering	6
49	International Conference on Software Engineering Advances	1
50	International Conference on Software Engineering and Data Mining	1
51	International Conference on Software Telecommunications and Computer Networks	1
52	International Conference on Theoretical Aspects of Software Engineering	1
53	International Conference on Wireless, Mobile and Sensor Networks	1
54	International Conference on World Wide Web	4
55	International Multi-Symposium on Computer and Computational Sciences	1
56	International Telecommunication Energy Conference	1
57	International Working Conference on Software Code Analysis and Manipulation	1
58	International Workshop on Automation of Software Test	1
59	International Workshop on Security Measurements and Metrics	1
60	MCETECH	1

(continued on next page)

Table 4 (*continued*)

No.	Title	No. of studies
61	Network and Distributed System Security Symposium	3
62	OTM Confederated International Conference on the Move to Meaningful Internet Systems	1
63	Pacific Rim International Symposium on Dependable Computing	1
64	RAID	1
65	SEUS	1
66	SPIN	1
67	Symposium on Library-Centric Software Design	1
68	TACAS	1
69	USENIX Conference on Offensive Technologies	1
70	USENIX Conference on Security	2
71	Web 2.0 Security and Privacy Workshop	1
72	Workshop on Programming Languages and Analysis for Security	2

Monitoring, taint-tracking and filtering are some of the dynamic analysis techniques currently employed to mitigate XSS attacks. However, attackers can still use obfuscation techniques to evade XSS filtering tools and inject JavaScript code.

The most ideal solution is to eliminate XSS vulnerabilities from the root cause, that is, the source code [124]. However, in real world web applications, obtaining the source code or implementing patches can be difficult. Hence, static analysis techniques are most useful during the application development and before deployment. Dynamic analysis techniques such as penetration testing technique can be used to exploit web applications during run time in order to determine if they are still vulnerable to XSS attacks.

The OWASP (Open Web Application Security Project) Foundation [3] has released an XSS prevention model, the XSS Prevention Cheat Sheet [130] that can be used for free as a guide to eliminate XSS vulnerabilities in web applications. The model provides specific rules to be followed when developing and testing applications. They also provide the Enterprise Security API (ESAPI), which is an open source security library that enables programmers incorporate security in their application development.

4.3. On which area(s) is research on XSS mostly focused?

The studies focused on two main areas: XSS attacks and XSS vulnerabilities. Sixty-nine studies (60%) focused on XSS attacks while 37 studies (32.2%) focused on XSS vulnerabilities. The remaining 9 studies (7.8%) combine more than one area. To understand the areas better, we divided each main area into categories namely, prevention, detection, and implementation for XSS attacks; detection, prevention, and prediction for XSS vulnerabilities.

As indicated in Table 8, the areas of focus of the studies are categorised into six groups. A seventh group is identified to capture the studies that focused in more than one area. XSS attack prevention is the highest category with a percentage of 43.5%, comprising almost half of the studies. This is not surprising as it is more desirable to prevent attacks than to provide contingency plans after the attacks have already occurred. A smaller number of studies (15) focused on detecting XSS attacks in running web applications, comprising 13.9% of the whole studies.

Four of the studies focused on attack implementation, demonstrating the possibility of XSS attacks on certain platforms and technologies. In [29], the possibility of spreading XSS worms and how fast it can be in social networks is demonstrated. Bencsath et at [18] shows how a form of XSS attacks, called cross channel scripting (XCS), can be mounted on embedded devices, and [34] demonstrated attacks through spoofing TCP/IP protocols. Kieyzum et al. [52] also implemented attacks on web sites by mutating inputs.

Detecting vulnerabilities in web applications is also an important area in XSS research with 32 of the studies (27.8%) focusing on it. It makes up 86.5% of the studies that focused mainly on XSS vulnerabilities. Detecting XSS vulnerabilities should be a first priority before an application is deployed, during implementation and testing. Vulnerability prevention and prediction received much less attention with only three [45,49,92] and two [89,90] studies focusing on them, respectively. However, these areas should be more focused on in XSS research as prevention is much better than cure.

The 9 studies under the combination category focused on more than one area each. Two studies focused on both vulnerability detection and attack prevention [42,75] and another couple of studies [17,88] focused on vulnerability detection and removal. The other 5studies [37,61,78,113,117] each focused on vulnerability detection and attack detection; attack detection and attack prevention; vulnerability prevention and attack prevention; vulnerability detection and attack implementation; and attack implementation and attack prevention, respectively.

4.4. Which of the three types of XSS is addressed the most?

Interestingly, 82(71.3%) out of the 115 studies did not specified what type of XSS their proposed solutions addressed. Some of them discussed the types of XSS but did not mention which type their solutions addressed. Out of the remaining 32 studies, 10 addressed reflected XSS (8.7%), 6 addressed stored XSS (5.2%), 1 addressed DOM-based XSS (0.9%), 3 addressed all three types (2.6%), and 13 addressed both reflected and stored XSS (11.3%).

Thus the type of XSS addressed the most in this review is reflected XSS. In total, 26 studies indicated to have provided a solution to reflected XSS. This is in line with literature where reflected XSS described as being the easiest type to detect. Stored XSS is the most dangerous of all types as the malicious scripts are stored in areas of the applications where whoever visits them gets attacked [1]. Some of the studies provide solutions for both reflected and stored XSS. Since both types affect the server side of the application, it is more feasible and easier to address them together than with DOM-based XSS.

Three of the studies addressed all the three types of XSS [22,80,97]. This means they have to address both the server side and client side. Although it requires more effort to address all the XSS types, it is a better initiative to have one solution that can address them all in an application.

Only one study [99] addressed DOM-based XSS. This can be attributed to the fact that DOM-based XSS is the least known type of XSS [1–3]. Some studies did not even include it in their descriptions of XSS type. However, it is a type of XSS worth noting. Unlike reflected and stored XSS, it exploits vulnerabilities in the client side script in the browser and not in the server side of an application. The inability to sometimes access client side scripts for analysis makes it more difficult to address this type of XSS, hence its low coverage.

Table 5
Summary of review findings.

Study	Summary of proposed technique/solution	Area of focus	Type of XSS
S001	A proxy program that imitates the human immune system by learning the behaviour of malicious data and uses the information to detect XSS attacks	Attack Detection	Not specified
S002	A methodology and tool that employs both Static Taint Analysis and Symbolic Code Execution to identify XSS and SQL injections vulnerabilities in PHP web applications	Vulnerability Detection	Not specified
S003	An algorithm that scans ASP.NET programs for the detection of cross site scripting and SQL injection security vulnerabilities	Vulnerability Detection	Not specified
S004	A detection tool that employs Forward and Backward Symbolic String Analysis to detect XSS, SQL injection, and malicious file execution in web applications	Attack Detection	Not specified
S005	A tool that evaluates URLs with the aim of identifying properties that can produce a valid JavaScript parse tree, which can lead to possible XSS exploits	Attack Detection	Not specified
S006	A search based approach, which integrates Static Taint Analysis, Genetic Algorithms, and Constraint solving to automatically generate test cases that will identify cross site scripting vulnerabilities in PHP applications	Vulnerability Detection	Reflected XSS
S007	A search based approach, which integrates Static Taint Analysis and Genetic Algorithms to automatically generate test cases that will identify cross site scripting vulnerabilities in PHP applications	Vulnerability Detection	Reflected XSS
S008	To address the problem of security oracle for cross site scripting by collecting HTML pages in safe conditions and use them to construct a safe model of the applications. The oracle is then used to detect attacks when an application displays a page not compliant with the safe model	Attack Detection	Reflected XSS
S009	An XML-based approach solution that uses the XML Schema Definition (XSD) to generate possible input part of a web page, which can later be used to validate future pages generated from user inputs. The method prevents untrusted user input from altering the structure of the code	Attack Detection	Stored XSS
S010	(1) A two-principled, security enhanced browser content-sniffing algorithm that helps to avoid privilege escalation and to use prefix-disjoint signatures to prevent content-sniffing XSS attacks (2) An upload filter based on models that protect web site from content-sniffing XSS attacks	Attack Prevention	Not specified
S011	A new design for a filter that can block scripts after HTML parsing but before it is executed	Attack Prevention	Reflected XSS
S012	A two-part algorithm that enables the detection and fixing of XSS vulnerabilities in Java web applications. The first part uses program slicing technique to identify the vulnerability and the second part uses program transformation to fix the vulnerability	Vulnerability Detection and removal	Not specified
S013	A framework that demonstrates how malicious code can be injected in web pages stored in embedded devices with networking capability and later used to launch XSS attacks with admin privileges when the devices are connected online during maintenance	Attack Implementation	Not specified
S014	XSS-GUARD: A framework that prevents XSS attacks on the server side by identifying and removing malicious scripts before any response page is generated for any HTML request	Attack Prevention	Not specified
S015	A browser extension that serves as client-side defence against cross channel scripting, a form of XSS that affects embedded devices by injecting malicious scripts through file transfer protocol, P2P networks, or file logs	Attack Prevention	Stored
S016	A model that helps to validate data entry at the application level of web services in order to prevent XSS and SQL injection	Attack Prevention	Not specified
S017	An approach that blocks the self-propagation of JavaScript worms through DOM access and unauthorized HTTP request, and prevents all forms of XSS worms in social network sites	Attack Prevention	All
S018	Rubyx: A tool that uses Symbolic Execution technique to test for security vulnerabilities, including XSS, in Ruby-on-Rails web applications	Attack Prevention	Not specified
S019	An automated vulnerability scanner that tests for XSS and SQL injection vulnerabilities based on the injection points of web applications using Black-box testing and Crawling technique	Vulnerability detection	Not specified
S020	An approach that enables the detection of XSS and SQL injection vulnerabilities through the use of N-Gram analysis to extract malicious code and Support Vector Machines (SVM) to classify it as XSS or SQL injection	Vulnerability detection	Not specified
S021	An ontology-based intrusion detection approach that is extended to include diagnostic features and is used in the detection of XSS and SQL injection attacks	Attack detection	Not specified
S022	An approach that uses both static and dynamic analysis of web applications to identify XSS vulnerabilities	Vulnerability detection	Not specified
S023	An approach that uses a combination of Model Inference and Evolutionary Fuzzing techniques to test application servers and detect reflected XSS vulnerabilities	Vulnerability detection	Reflected
S024	A general model derived through simulating the propagation behaviour of XSS worms in social networks that can be used to predict how fast XSS worms can spread on social networks	Attack implementation	Stored XSS
S025	An approach that can serve as a benchmark to evaluate and compare XSS and SQL injection vulnerability scanners in order to assess their capabilities and limitations	Vulnerability detection	Not specified
S026	An Intrusion Detection System for XSS that captures potential client side executable content and its hashing, and later reprocessed for any difference that will indicate XSS attack	Attack detection	Not specified
S027	A multi-agent scanner that automatically scans web sites for the presence of stored XSS vulnerabilities	Vulnerability detection	Stored XSS
S028	An approach to prevent XSS attacks through the use of X.509 certificates and XACML for the expression of authorization policies	Attack prevention	Reflected and Stored XSS
S029	An approach to demonstrate the possibility of conducting security attacks including off-path injection XSS attacks through spoofing the TCP/IP protocol	Attack Implementation	Not specified
S030	An approach to use type systems to automatically enforce programming guidelines that prevent XSS attacks in Java programs	Attack prevention	Not specified
S031	Noncespaces: A technique that uses randomized XML namespaces to enable the server identify untrusted content and the client can use the information to enforce policies that will prevent XSS attacks	Attack prevention	Not specified
S032	An illustration of the possibility of carrying out XSS attacks with Scalable Vector Graphics (SVG) images through the use of tags, and an approach to limiting the risks of such attacks in web applications by removing the malicious content from a SVG file	Attack implementation and prevention	Not specified
S033	An approach to detect SQL injection and XSS attacks by implementing a security aspect through the use of Aspect-Oriented Programming (AOP) framework that validates and filters user input	Attack detection	Not specified
S034	A server-side solution to XSS and SQL injection attacks that uses MD5 algorithm and grammar expression rules to detect the attacks using a reverse proxy	Attack detection	Not specified
S035	A precise analysis of XSS sanitizers' behaviour using BEK language that enables to write and analyse string	Attack prevention	Not specified

(*continued on next page*)

Table 5 (*continued*)

Study	Summary of proposed technique/solution	Area of focus	Type of XSS
	manipulation routines and compile them to general purpose languages such as JavaScript		
S036	A scheme that uses Bing-Value as HTTP response header in the browser, a binding mechanism that prevents XSS attacks	Attack prevention	Not specified
S037	A system that uses a proxy approach to detect and collect XSS vulnerabilities and uses the information to prevent XSS attacks	Vulnerability detection and attack prevention	Not specified
S038	A server side solution that uses signature based model to detect XSS attacks	Attack detection	Not specified
S039	SessionSafe: A combination of three server side techniques that helps to prevent session hijacking attacks, which are threats resulting from XSS vulnerabilities	Attack prevention	Not specified
S040	An approach that enforces secure generation for programming languages and prevents the creation of string-based injection vulnerabilities including XSS	Vulnerability prevention	Not specified
S041	A server side detection system for XSS attacks that detects reflected XSS attacks and discovers stored XSS by monitoring the application's HTTP traffic	Attack detection	Reflected and stored
S042	Pixy: A tool that statically tests PHP web application source code to detect XSS vulnerabilities by using flow-sensitive, interprocedural, and context-sensitive data flow analysis techniques	Vulnerability detection	Not specified
S043	A novel precise alias analysis targeted at the unique reference semantics commonly found in scripting languages that is able to detect many vulnerabilities including XSS	Vulnerability detection	Not specified
S044	A library that helps to enforce secure coding of database web applications using libraries and object models.	Vulnerability Prevention	Not specified
S045	Secubat: A security scanner that automatically checks web applications for XSS and SQL injection vulnerabilities by executing some form of attacks	Vulnerability detection	Reflected
S046	A gateway solution that uses web page classification, referrer string, and cookies techniques and is deployed at the front end of web applications to prevent reflected XSS and cross site request forgery attacks	Attack prevention	Reflected
S047	An automated technique that finds XSS and SQL injection vulnerabilities in web sites. The technique generates sample inputs, tracks taints through execution, and mutates inputs to produces exploits	Attack implementation	Reflected and stored
S048	A client-side solution to mitigate XSS attacks that acts as proxy and uses both manual and automatically generated rules	Attack prevention	Reflected and stored
S049	Noxes: A client-side solution that acts as a proxy and uses both manual and automatically generated rules to block XSS attacks by preventing information leakage from the user environment	Attack prevention	Reflected and stored
S050	An approach that uses Machine Learning techniques to learn the patterns of existing malicious codes and uses the information to classify new code as either malicious or not, thereby identifying XSS and SQL injection vulnerabilities	Vulnerability detection	Not specified
S051	A solution that uses Java source code model checker, Bandera, to determine if secure programming guidelines are followed, and checks for XSS and SQL injection vulnerabilities	Vulnerability detection	Not specified
S052	An approach called Perturbation-based Interactive UIV Testing (PIUIVT) that improves the effectiveness of vulnerability scanners for user-input-validation (UIV) testing for web applications by generating test inputs the reveal XSS and other vulnerabilities	Vulnerability detection	Not specified
S053	FIRM: A system that embeds inline reference monitor (IRM) in web pages hosting Flash content and protects it though controlling DOM methods and randomizing variables with sensitive data in order to prevent security attack including XSS	Attack prevention	Not specified
S054	A scheme that helps to eliminate injection vulnerabilities including XSS in applications built on AJAX frameworks by refining the same-origin policy in the browser, thus preventing attacks	Attack prevention	Not specified
S055	BLUEPRINT: A defense strategy that seeks to minimize the trust put on browsers for interpreting untrusted content by eliminating any dependence on the browser's parser	Attack prevention	Not specified
S056	A system that automatically generates attacks that exploit taint-based vulnerabilities including XSS in large Java web applications by using concrete model checking, dynamic monitoring, and program analysis techniques	Vulnerability detection and Attack implementation	Not specified
S057	An automated black-box vulnerability scanner that can find reflected and stored XSS in web applications by increasing testing depth and breadth, and using stateful fuzzing	Vulnerability detection	Reflected and stored
S058	A static string analyser for PHP that detects XSS vulnerabilities in PHP programs using a context-free grammar to approximate web pages generated by a program	Vulnerability detection	Not specified
S059	DESERVE: A monitor embedding framework that identifies exploitable statements in a source code using static backward slicing and embeds and helps to identify attacks including XSS	Attack detection	Reflected and stored
S060	An approach that improves dynamic tainting technique with character coding and complement aware components to protect applications against stored XSS attacks	Attack prevention	Stored
S061	A client–server architecture that enforces document structure integrity by combining randomization of web application code and runtime tracking of untrusted data to prevent reflected XSS attacks	Attack prevention	Reflected
S062	A dynamic checking compiler that automatically adds checks into web applications used in three-tier Internet services to prevent attacks including XSS by using taint analysis and HTML parsers	Attack prevention	Not specified
S063	A fully automated approach that is based on precisely tracking taintedness of data and checking specifically for dangerous content only in untrustworthy sources thereby preventing XSS attacks and others	Attack prevention	Not specified
S064	SessionShield: A lightweight protection mechanism against a form of XSS attack called session hijacking, which detects session identifiers in incoming HTTP traffic and isolates them from the browser thereby preventing attacks	Attack prevention	Not specified
S065	Using features from web document and ULR to classify patterns of cross site scripting attacks by employing machine learning techniques	Attack detection	Not specified
S066	A browser-based defense mechanism against reflected XSS attacks that uses approximate string matching to detect reflected content	Attack prevention	Reflected
S067	LAPSE: An Eclipse plugin that analyses Java EE applications for the detection of security vulnerabilities including XSS	Vulnerability detection	Not specified
S068	A solution to identify categories of programming flaws leading software bugs and indexing existing vulnerability reports against those categories	Vulnerability detection	Not specified
S069	A method to control JavaScript execution by preventing or modifying inappropriate behaviour such as malicious injected scripts, thereby preventing XSS attacks	Attack prevention	Not specified

Please cite this article in press as: I. Hydara et al., Current state of research on cross-site scripting (XSS) – A systematic literature review, Inform. Softw. Technol. (2014), http://dx.doi.org/10.1016/j.infsof.2014.07.010

Table 5 (*continued*)

Study	Summary of proposed technique/solution	Area of focus	Type of XSS
S070	An intrusion detection system that identifies vulnerabilities including XSS and prevents attacks on such vulnerabilities using inter server communication techniques	Vulnerability detection and attack prevention	Not specified
S071	A new technique called Dynamic Cookies Rewriting that renders cookies useless for cross site scripting attacks	Attack prevention	Reflected and stored
S072	SCRIPTGUARD: An automatic context-sensitive sanitizer for ASP.NET applications that can detect and repair incorrect placement of sanitizers thus mitigating XSS and XCS attacks in legacy code	Attack prevention	Not specified
S073	Prevention of website exploitation of cross site scripting and SQL injection vulnerabilities in PHP source code based on automated data type detection of input parameters, using a new tool: IPAAS (Input Parameter Analysis System)	Vulnerability prevention and Attack prevention	Not specified
S074	An approach that injects comments for legitimate JavaScript code that encode legitimate code features in terms of method definition and call signatures, which makes it difficult to inject legitimate code thereby preventing injection attacks such as XSS	Attack prevention	Not specified
S075	An approach that employs mutation-based testing technique to generate adequate test data to test for XSS vulnerabilities in PHP applications	Vulnerability detection	All
S076	To propose an automated framework to detect cross site scripting attacks at the server side based on boundary injection and policy generation	Attack detection	Not specified
S077	A trustworthiness testing approach of suspected phishing web sites based on behaviour model that uses Finite State Machine techniques to determine if a website can be trusted, which can detect XSS attacks	Attack detection	Not specified
S078	A language independent solution to block XSS attacks using the Service-Oriented Architecture approach	Attack prevention	Not specified
S079	A behaviour-based anomaly detection approach that puts a security layer on top of the web application to prevent XSS attacks	Attack prevention	Not specified
S080	A thread-based solution for efficient process utilization of the web server and to prevent XSS attacks on AJAX applications	Attack prevention	Not specified
S081	An approach for the thorough auditing of source code to defend against XSS attacks by extracting implemented defences in the code and check them for adequacy and potential risks	Attack prevention	Not specified
S082	A proposed method that will recover the defence model implemented in program source code and to check the model against attacks based on given guidelines	Attack prevention	Reflected and stored
S083	To detect and remove the XSS vulnerabilities web applications using static analysis and pattern matching techniques	Vulnerability detection and removal	Reflected and stored
S084	To classify various input sanitization methods into different types and use code attributes to represent the types. Then employ data mining techniques to predict SQL injection and cross site scripting	Vulnerability prediction	Not specified
S085	An approach to predicting XSS and SQL injection vulnerabilities using input validation and input sanitization patterns	Vulnerability prediction	Not specified
S086	An integrated model to prevent reflected XSS and SQL injection attacks in PHP web applications	Attack prevention	Reflected
S087	The construction of a common XSS vulnerability enumeration that can help security practitioners recognise common developer patterns leading to coding errors in PHP	Vulnerability Prevention	Not specified
S088	A black-box analysis methodology for public Cloud interfaces that provides countermeasures for XSS attacks	Attack prevention	Not specified
S089	A framework for the evaluation of web intrusion prevention systems	Attack prevention	Not specified
S090	An approach to preventing the propagation of XSS worms by monitoring outgoing request that send self-replicating payloads	Vulnerability detection	Not specified
S091	A model checking method that uses the automatic modelling algorithm for the HTML code to defend against XSS attacks	Attack prevention	Not specified
S092	A hybrid client–server solution that combines the benefits of both server and client-side protection mechanisms to mitigate XSS attacks using anomaly detection and control flow analysis	Attack prevention	All
S093	A protection scheme against attacks deployed by hiding the violation of the same origin policy including XSS that finds mismatches between the origin and target pages of HTTP request	Attack prevention	Reflected and stored
S094	Alhambra: A browser-based system for testing enforcing security policies to prevent XSS attacks using taint-tracking engine and browsing history	Attack prevention	DOM-based
S095	A Webmail XSS fuzzer, which works on a lexical based mutation engine and helps to discover XSS vulnerabilities in webmail applications	Vulnerability detection	Not specified
S096	A client-side solution that uses step-by-step approach to protect web applications against XSS attacks	Attack prevention	Not specified
S097	A client-side solution that uses a step by step approach to detect XSS attacks	Attack prevention	Not specified
S098	An optimum tuning method based on the application firewall that uses keyword filtering and re-treatment to effectively block assaults including XSS attacks	Attack prevention	Not specified
S099	BIXAN: A browser independent XSS sanitizer that uses a JavaScript tester, a HTML parser, and identification of static tags to prevent XSS attacks	Attack prevention	Not specified
S100	Noncespaces: A technique that enables web clients to distinguish between trusted and untrusted content to prevent exploitation of XSS vulnerabilities	Attack prevention	Reflected and stored
S101	FlashOver: A system that automatically scans Rich Internet Applications for XSS vulnerabilities by using a combination of static and dynamic code analysis techniques	Vulnerability detection	Not specified
S102	WebAppArmor: A framework that incorporates techniques based on static and dynamic analysis, symbolic evaluation and execution monitoring to prevent XSS and other attacks on existing web applications	Attack prevention	Not specified
S103	A client-side solution that uses dynamic data tainting and static analysis to prevent XSS attacks	Attack prevention	Not specified
S104	A scheme on how to collect evidence after XSS attacks and strategies to prevent XSS attacks	Attack prevention	Not specified
S105	A static stored XSS detection algorithm integrated with program slicing method to detect stored XSS vulnerabilities	Vulnerability detection	Stored
S106	A static analysis for finding cross site scripting vulnerabilities that addresses weak or absent input validation by combining tainted information flow with string analysis	Vulnerability detection	Reflected and stored
S107	A study of the security of XSS sanitization abstractions provided by frameworks that shows the gap between the abstractions and the application requirements	Vulnerability detection	Not specified
S108	SWAP: A server-side solution for detecting and preventing XSS attacks using a reverse proxy that intercepts all HTML responses	Attack Detection and Attack Prevention	Not specified
S109	A static analysis tool to detect XSS attacks and SQL injection vulnerabilities on ASP programs based on taint analysis	Vulnerability detection	Not specified

(*continued on next page*)

Please cite this article in press as: I. Hydara et al., Current state of research on cross-site scripting (XSS) – A systematic literature review, Inform. Softw. Technol. (2014), http://dx.doi.org/10.1016/j.infsof.2014.07.010

Table 5 (*continued*)

Study	Summary of proposed technique/solution	Area of focus	Type of XSS
S110	A model-based penetration testing approach for web applications that uses TTCN-3 technique for test case generation related to XSS	Vulnerability detection	Not specified
S111	STRANGER: An automata-based string analysis tool for finding and eliminating string-related vulnerabilities including XSS in PHP applications	Vulnerability detection	Not specified
S112	A set of sound abstractions for strings and string operations that allow for both efficient and precise verification of string manipulating programs to show absence of vulnerabilities	Vulnerability Detection and Attack Detection	Not specified
S113	D-WAV: A web application vulnerability detection tool that uses characteristics of web forms to detect vulnerabilities including XSS	Vulnerability detection	Not specified
S114	An execution-flow analysis technique is proposed that analyses the execution flow of the client-side JavaScript before the requested arrives at the browser	Attack detection	Not specified
S115	MBDS: A model-based, client-side system that automatically detects XSS vulnerabilities using both primitive and advanced models	Vulnerability detection	Not specified

Table 6
Summary of techniques/solutions provided by the studies.

Techniques/solutions	References	No. of studies	Percentage (%)
Static analysis	[7–12,17,23,25,32,47,48,55,57,63,64,70,72,77,78,80,88,110,111,114,116,117]	27	23.5
Dynamic analysis	[6,14–16,18–20,22,24,26,31,33,34,36–42,46,50–54,58–62,65–69,71,75,76,79,81,83,84,93,95,97–105,113,118,119]	57	49.6
Static and dynamic analysis	[27,74,106–108]	5	4.3
Secure programming	[35,45,49]	3	2.6
Modelling	[13,21,28,29,43,44,56,82,85–87,89–91,96,112,115,120]	18	15.7
Others	[30,73,92,94,109]	5	4.3

Table 7
Comparison between dates and proposed techniques/solutions.

Year	Static analysis	Dynamic analysis	static and dynamic analysis	Secure programming	Modelling	Others
2012	√√√√√√	√√√√√√√	√	√	√√√√√√√√√	
2011	√√√√√√√√√	√√√√√√√√√√√√√√√			√	√
2010	√√√√√√√	√√√√√√	√	√	√√	
2009	√	√√√√√√√√√√√√	√		√√	
2008	√	√√√√√√				
2007		√√√√√√√	√	√	√√√	√√√
2006	√√	√√			√	
2005	√	√				√
2004		√	√			

Table 8
Areas focused on by the studies and their percentage.

Area of focus	No. of studies	Percentage (%)
Attack detection	15	13
Attack prevention	50	43.5
Attack implementation	4	3.5
Vulnerability detection	32	27.8
Vulnerability prevention	3	2.6
Vulnerability prediction	2	1.7
Combinations	9	7.8

4.5. Limitations of this study

Despite the fact that we try our best to adhere to the guidelines by Kitchenham [4], we can still identify some limitations to our study. Our search of relevant studies may have not been thorough, although we have checked many online databases as well as checked references of some articles. Thus, we could have missed some important and relevant studies. We limited the time span of the studies only to those published up to end of 2012. Therefore articles that were published from 2013 onwards were not included in our review. Also, we limited the review to academic studies only.

Table 9
Categories of XSS addressed by the studies and their percentage.

Type of XSS	No. of studies	Percentage (%)
Reflected XSS	10	8.7
Stored XSS	6	5.2
DOM-based XSS	1	0.9
All	3	2.6
Reflected and Stored XSS	13	11.3
Not specified	82	71.3

5. Conclusion

We have conducted a systematic literature review of 115 articles related to research on XSS. We have identified the solutions/techniques proposed in the studies, the areas the studies focused on and the types of XSS the solutions/techniques addressed. The proposed solutions are many and diverse. Most of them are focused on preventing XSS attacks and detecting vulnerabilities. Only two studies have discussed the removal of XSS vulnerabilities from the source code. This is an important aspect of XSS research, as the absence of vulnerabilities will prevent attacks from occurring and save resources.

Many research activities have been conducted to address problems related to XSS since their discovery. Despite all the efforts

Please cite this article in press as: I. Hydara et al., Current state of research on cross-site scripting (XSS) – A systematic literature review, Inform. Softw. Technol. (2014), http://dx.doi.org/10.1016/j.infsof.2014.07.010

over the years to eliminate them, XSS vulnerabilities are still prevalent in web application source codes and attacks are still taking place victimizing site owners and innocent users. Security should be addressed at every phase of web application development and throughout the application lifecycle. Perhaps, it is time for security researchers and developers to start focusing more on eliminating XSS vulnerabilities from web application source codes before deployment. More work is needed to develop policies and tools, such as the OWASP projects, that will enforce the development of secure applications.

Since it is quite impossible to eliminate all XSS vulnerabilities before deployment of an application, penetration testing and other dynamic analysis techniques should be used after deployment to continually test the application. This will ensure more protection from attackers and reduce XSS incidents.

Acknowledgement

This work was supported by the Malaysian Ministry of Education under the Fundamental Research Grant Scheme (FRGS 08-02-13-1362).

References

[1] S. Fogie, J. Grossman, R. Hansen, A. Rager, P.D Petkov, XSS Attacks: Cross Site Scripting Exploits and Defense, Elsevier, Inc./Syngress Publishing, Inc., Burlington, MA, 2007. p. 480.

[2] CWE, CWE – CWE-79: Improper Neutralization of Input During Web Page Generation ('Cross-site Scripting') (2.5), The MITRE Corporation [Online]. <http://cwe.mitre.org/data/definitions/79.html>.

[3] OWASP, Cross-site Scripting (XSS) – OWASP, OWASP. [Online]. <https://www.owasp.org/index.php/Cross-site_Scripting_(XSS)>.

[4] S. Kitchenham, B.A. Charters, Guidelines for performing systematic literature reviews in software engineering, Keele (2007).

[5] The Mendeley Support Team, Free reference manager and PDF organizer|Mendeley [Online]. <http://www.mendeley.com/>.

[6] E. Adi, A design of a proxy inspired from human immune system to detect SQL injection and cross-site scripting, Procedia Eng. 50 (Icasce) (2012) 19–28. January 2012.

[7] G. Agosta, A. Barenghi, A. Parata, G. Pelosi, Automated Security Analysis of Dynamic Web Applications through Symbolic Code Execution, in: 2012 Ninth International COnference on Information Technology – New Generations, 2012, pp. 189–194.

[8] H. Al-amro, E. El-qawasmeh, Discovering Security Vulnerabilities And Leaks In ASP. NET Websites, in: Cyber Security, Cyber Warfare and Digital Forensic (CyberSec), 2012 International Conference on, 2012, pp. 329–333.

[9] D. Arulsuju, Hunting Malicious Attacks in Social Networks, in: Advanced Computing (ICoAC), 2011 Third International Conference on, 2011, pp. 13–17.

[10] E. Athanasopoulos, A. Krithinakis, E.P. Markatos, Hunting Cross-Site Scripting Attacks in the Network, in: W2SP 2010: Web 2.0 Security and Privacy Workshop, 2010.

[11] A. Avancini, M. Ceccato, Towards Security Testing with Taint Analysis and Genetic Algorithms, in: Proceedings of the 2010 ICSE Workshop on Software Engineering for Secure Systems, 2010, no. Section 5, pp. 65–71.

[12] A. Avancini, M. Ceccato, Security Testing of Web Applications: A Search-Based Approach for Cross-Site Scripting Vulnerabilities, in: 2011 IEEE 11th International Working Conference on Source Code Analysis and Manipulation, 2011, pp. 85–94.

[13] A. Avancini, M. Ceccato, F.B. Kessler, Grammar Based Oracle for Security Testing of Web Applications, in: 2012 7th International Workshop on Automation of Software Test (AST), 2012, no. line 13, pp. 15–21.

[14] T.S. Barhoom, S.N. Kohail, A new server-side solution for detecting cross site scripting attack, Int. J. Comput. Inf. Syst. 3 (2) (2011) 19–23.

[15] A. Barth, J. Caballero, D. Song, Secure Content Sniffing for Web Browsers, or How to Stop Papers from Reviewing Themselves, in: 2009 30th IEEE Symposium on Security and Privacy, 2009, pp. 360–371.

[16] D. Bates, A. Barth, C. Jackson, Regular Expressions Considered Harmful in Client-side XSS Filters, in: Proc. 19th Int. Conf. World wide web – WWW '10, 2010, p. 91.

[17] P. Bathia, B.R. Beerelli, M. Laverdière, Assisting Programmers Resolving Vulnerabilities in Java Web Applications, in: CCIST 2011: Communications in Computer and Information Science, 2011, vol. 133, no. 1, pp. 268–279.

[18] B. Bencsath, L. Buttyan, T. Paulik, XCS Based Hidden Firmware Modification on Embedded Devices, in: SoftCOM 2011 19th International Conference on Software Telecommunications and Computer Networks, 2011, pp. 1–5.

[19] P. Bisht, V.N. Venkatakrishnan, XSS-GUARD: Precise dynamic prevention of cross-site scripting attacks, DIMVA 2008, Lect. Notes Comput. Sci., vol. 5137, 2008, pp. 23–43.

[20] H. Bojinov, E. Bursztein, D. Boneh, XCS: Cross Channel Scripting and its Impact on Web Applications, in: CCS '09: Proceedings of the 16th ACM Conference on Computer and Communications Security, 2009, pp. 420–431.

[21] R.B. Brinhosa, C.M. Westphall, C.B. Westphall, Proposal and Development of the Web Services Input Validation Model, in: 2012 IEEE Network Operations and Management Symposium (NOMS), 2012, pp. 643–646.

[22] Y. Cao, V. Yegneswaran, P. Porras, Y. Chen, POSTER : A Path-Cutting Approach to Blocking XSS Worms in Social Web Networks, in: CCS '11: Proceedings of the 18th ACM Conference on Computer and Communications Security, 2011, pp. 745–747.

[23] A. Chaudhuri, J.S. Foster, Symbolic security analysis of ruby-on-rails web applications, in: Proc. 17th ACM Conf. Comput. Commun. Secur. – CCS '10, 2010, p. 585.

[24] J.-M.C.J.-M. Chen, C.-L.W.C.-L. Wu, An automated vulnerability scanner for injection attack based on injection point, in: 2010 International Computer Symposium ICS2010, 2010, pp. 113–118.

[25] J. Choi, H. Kim, C. Choi, P. Kim, Efficient Malicious Code Detection Using N-Gram Analysis and SVM, in: 2011 14th International Conference on Network Based Information Systems, 2011, pp. 618–621.

[26] L. Coppolino, S.D. Antonio, I.A. Elia, L. Romano, From intrusion detection to intrusion detection and diagnosis: an ontology-based approach, SEUS 2009, Lect. Notes Comput. Sci., vol. 5860, 2009, pp. 192–202.

[27] G.A. Di Lucca, A.R. Fasolino, M. Mastroianni, P. Tramontana, Identifying Cross Site Scripting Vulnerabilities in Web Applications, in: 26th Annual International Telecommunications Energy Conference, 2004, pp. 71–80.

[28] F. Duchene, R. Groz, S. Rawat, J.-L. Richier, XSS Vulnerability Detection Using Model Inference Assisted Evolutionary Fuzzing, in: 2012 IEEE Fifth International Conference on Software Testing, Verification and Validation, 2012, no. Itea 2, pp. 815–817.

[29] M.R. Faghani, H. Saidi, Social Networks' XSS Worms, in: 2009 International Conference on Computational Science and Engineering, 2009, pp. 1137–1141.

[30] J. Fonseca, M. Vieira, H. Madeira, Testing and Comparing Web Vulnerability Scanning Tools for SQL Injection and XSS Attacks, in: 13th Pacific Rim Int. Symp. Dependable Comput. (PRDC 2007), December 2007, pp. 365–372.

[31] C.M. Frenz, J.P. Yoon, XSSmon: A Perl Based IDS for the Detection of Potential XSS Attacks, in: 2012 IEEE Long Island Systems, Application and Technology Conference (LISAT), 2012, pp. 1–4.

[32] E. Galan, A. Alcaide, A. Orfila, J. Blasco, A Multi-agent Scanner to Detect Stored-XSS Vulnerabilities, in: 2010 International Conference for Internet Technology and Secured Transactions (ICITST), 2010, pp. 1–6.

[33] J. Garcia-alfaro, G. Navarro-arribas, Prevention of cross-site scripting attacks on current web applications, OTM 2007, Lect. Notes Comput. Sci., vol. 4804, 2007, pp. 1770–1784.

[34] Y. Gilad, A. Herzberg, Off-Path Attacking the Web, in: Proceedings of the 6th USENIX conference on Offensive Technologies, 2012, pp. 1–12.

[35] R. Grabowski, M. Hofmann, K. Li, Type-based enforcement of secure programming guidelines—code injection prevention at SAP, FAST 2011, Lect. Notes Comput. Sci., vol. 7140, 2012, pp. 182–197.

[36] M. Van Gundy, H. Chen, Noncespaces : Using Randomization to Enforce Information Flow Tracking and Thwart Cross-Site Scripting Attacks, in: 16th Annual Network and Distributed System Security Symposium Proceedings, NDSS Symposium 2009, 2009.

[37] M. Heiderich, T. Holz, Crouching Tiger – Hidden Payload : Security Risks of Scalable Vectors Graphics, in: CCS '11: Proceedings of the 18th ACM Conference on Computer and Communications Security, 2011, pp. 239–250.

[38] G. Hermosillo, R. Gomez, L. Seinturier, L. Duchien, AProSec: An Aspect for Programming Secure Web Applications, in: Second International Conference on Availability, Reliability and Security (ARES'07), 2007, no. 1, pp. 1026–1033.

[39] S.F. Hidhaya, A. Geetha, Intrusion protection against SQL injection and cross site scripting attacks using a reverse proxy, Commun. Comput. Informat. Sci. 335 (2012) 252–263.

[40] P. Hooimeijer, B. Livshits, D. Molnar, P. Saxena, and M. Veanes, "Fast and Precise Sanitizer Analysis with BEK", in: SEC'11: Proceedings of the 20th USENIX conference on Security, 2011, pp. 1–16.

[41] G. Iha, H. Doi, An Implementation of the Binding Mechanism in the Web Browser for Preventing XSS Attacks: Introducing the Bind-Value Headers, in: 2009 International Conference on Availability Reliability and Security, 2009, pp. 966–971.

[42] O. Ismail, M. Etoh, Y. Kadobayashi, S. Yamaguchi, A Proposal and Implementation of Automatic Detection/Collection System for Cross-Site Scripting Vulnerability, in: 18th International Conference on Advanced Information Networking and Applications, 2004. AINA 2004, 2004, pp. 145–151.

[43] S. Jayamsakthi, M. Ponnavaikko, Risk mitigation for cross site scripting attacks using signature based model on the server side, Second Int. Multi-Symposiums Comput. Comput. Sci. (IMSCCS 2007), August 2007, pp. 398–405.

[44] M. Johns, SessionSafe: implementing XSS immune session handling, Lect. Notes Comput. Sci., vol. 4189, 2006, pp. 444–460.

[45] M. Johns, C. Beyerlein, R. Giesecke, J. Posegga, Secure code generation for web applications, Lect. Notes Comput. Sci. 5965 (2010) 96–113.

[46] M. Johns, B. Engelmann, J. Posegga, XSSDS: Server-side Detection of Cross-site Scripting Attacks, in: 2008 Annual Computer Security Applications Conference, 2008, pp. 335–344.

[47] N. Jovanovic, C. Kruegel, E. Kirda, Pixy: A Static Analysis Tool for Detecting Web Application Vulnerabilities (Short Paper), in: Proceedings of the 2006 IEEE Symposium on Security and Privacy, 2006.

[48] N. Jovanovic, C. Kruegel, E. Kirda, Precise Alias Analysis for Static Detection of Web Application Vulnerabilities, in: PLAS '06: Proceedings of the 2006 workshop on Programming languages and analysis for security, 2006, pp. 27–36.

[49] N. Juillerat, Enforcing Code Security in Database Web Applications using Libraries and Object Models, in: Proc. 2007 Symp. Libr. Softw. Des. – LCSD '07, 2007, pp. 31–41.

[50] S. Kals, E. Kirda, C. Kruegel, N. Jovanovic, SecuBat: A Web Vulnerability Scanner, in: WWW '06: Proceedings of the 15th international conference on World Wide Web, 2006, pp. 247–256.

[51] F. Kerschbaum, Simple Cross-site Attack Prevention, 2007 Third Int. Conf. Secur. Priv. Commun. Networks Work. – Secur. 2007, 2007, pp. 464–472.

[52] A. Kieyzun, P.J. Guo, K. Jayaraman, M.D. Ernst, Automatic creation of SQL Injection and cross-site scripting attacks, in: 2009 IEEE 31st International Conference on Software Engineering, 2009, pp. 199–209.

[53] E. Kirda, N. Jovanovic, C. Kruegel, G. Vigna, Client-side cross-site scripting protection, Comput. Secur. 28 (7) (2009) 592–604. October.

[54] E. Kirda, C. Kruegel, G. Vigna, N. Jovanovic, Noxes: A Client-Side Solution for Mitigating Cross-Site Scripting Attacks, in: SAC '06: Proceedings of the 2006 ACM symposium on Applied computing, 2006, pp. 330–337.

[55] R. Komiya, I. Paik, M. Hisada, Classification of malicious web code by machine learning, in: 2011 3rd International Conference on Awareness Science and Technology iCAST, 2011, pp. 406–411.

[56] K.L.K. Li, Towards Security Vulnerability Detection by Source Code Model Checking, in: Software Testing Verification and Validation Workshops ICSTW 2010 Third International Conference on, 2010, pp. 381–387.

[57] N. Li, T. Xie, M. Jin, C. Liu, Perturbation-based user-input-validation testing of web applications, J. Syst. Softw. 83 (11) (Nov. 2010) 2263–2274.

[58] Z. Li, X. Wang, FIRM: Capability-based Inline Mediation of Flash Behaviors, in: ACSAC '10: Proceedings of the 26th Annual Computer Security Applications Conference, 2010, pp. 181–190.

[59] B. Livshits, Ú. Erlingsson, Using web application construction frameworks to protect against code injection attacks, in: Proc. 2007 Work. Program. Lang. Anal. Secur. – PLAS '07, 2007, p. 95.

[60] M. Ter Louw, V.N. Venkatakrishnan, Blueprint: Robust Prevention of Cross-site Scripting Attacks for Existing Browsers, in: 2009 30th IEEE Symposium on Security and Privacy, 2009, pp. 331–346.

[61] M. Martin, M.S. Lam, Automatic Generation of XSS and SQL Injection Attacks with Goal-Directed Model Checking, in: 17th Conference on Security Symposium, 2008, pp. 31–43.

[62] S. Mcallister, E. Kirda, C. Kruegel, Leveraging user interactions for in-depth testing of web applications, Lect. Notes Comput. Sci. 5230 (2008) 191–210.

[63] Y. Minamide, Static Approximation of Dynamically Generated Web Pages, in: WWW '05: Proceedings of the 14th International Conference on World Wide Web, 2005, pp. 432–441.

[64] A. Mohosina, M. Zulkernine, DESERVE: A Framework for Detecting Program Security Vulnerability Exploitations, in: 2012 IEEE Sixth International Conference on Software Security and Reliability, 2012, pp. 98–107.

[65] R. Mui, P. Frankl, Preventing web application injections with complementary character coding, Lect. Notes Comput. Sci. 6879 (2011) 80–99.

[66] Y. Nadji, P. Saxena, D. Song, Document Structure Integrity : A Robust Basis for Cross-site Scripting Defense, in: 16th Annual Network and Distributed System Security Symposium Proceedings, NDSS Symposium 2009, 2009.

[67] S. Nanda, L.-C. Lam, T. Chiueh, Dynamic multi-process information flow tracking for web application security, in: Proc. 8th ACM/IFIP/USENIX Int. Conf. Middlew. – Middlew. '07, 2007, p. 1.

[68] A. Nguyen-tuong, S. Guarnieri, D. Greene, J. Shirley, D. Evans, Automatically hardening web applications using precise tainting, IFIP Adv. Inf. Commun. Technol. 181 (2005) 295–307.

[69] N. Nikiforakis, W. Meert, Y. Younan, M. Johns, W. Joosen, SessionShield: Lightweight Protection against Session Hijacking, in: ESSoS'11: Proceedings of the Third International Conference on Engineering Secure Software and Systems, 2011, pp. 87–100.

[70] A.E. Nunan, E. Souto, E.M. Santos, E. Feitosa, Automatic Classification of Cross-Site Scripting in Web Pages Using Document-based and URL-based Features, in: Computers and Communications (ISCC), 2012 IEEE Symposium on, 2012, pp. 702–707.

[71] R. Pelizzi, R. Sekar, Protection, usability and improvements in reflected XSS filters, in: Proceedings of the 7th ACM Symposium on Information, Computer and Communications Security – ASIACCS '12, 2012, p. 5.

[72] P.M. Pérez, J. Filipiak, J.M. Sierra, LAPSE + static analysis security software: vulnerabilities detection in Java EE applications, Commun. Comput. Inf. Sci. 184 (2011) 148–156.

[73] K. Petkov, Overcoming Programming Flaws : Indexing of Common Software Vulnerabilities, in: InfoSecCD '05: Proceedings of the 2nd Annual Conference on Information Security Curriculum Development, 2005, pp. 127–134.

[74] P.H. Phung, D. Sands, A. Chudnov, Lightweight Self-Protecting JavaScript, in: ASIACCS '09: Proceedings of the 4th International Symposium on Information, Computer, and Communications Security, 2009, pp. 47–60.

[75] R. Priyadarshini, D. Jagadiswaree, A. Fareedha, M. Janarthanan, A cross platform intrusion detection system using inter server communication technique, in: 2011 International Conference on Recent Trends in Information Technology ICRTIT, 2011, pp. 1259–1264.

[76] R. Putthacharoen, P. Bunyatnoparat, Protecting cookies from Cross Site Script attacks using Dynamic Cookies Rewriting technique, in: 13th International Conference on Advanced Communication Technology ICACT2011, 2011, pp. 1090–1094.

[77] P. Saxena, D. Molnar, B. Livshits, ScriptGard : Automatic Context-Sensitive Sanitization for Large-Scale Legacy Web Applications Categories and Subject Descriptors, in: CCS '11: Proceedings of the 18th ACM conference on Computer and Communications Security, 2011, pp. 601–614.

[78] T. Scholte, W. Robertson, D. Balzarotti, E. Kirda, Preventing Input Validation Vulnerabilities in Web Applications through Automated Type Analysis, in: 2012 IEEE 36th Annual Computer Software and Applications Conference, 2012, pp. 233–243.

[79] H. Shahriar, M. Zulkernine, Injecting Comments to Detect JavaScript Code Injection Attacks, in: 2011 IEEE 35th Annual Computer Software and Applications Conference Workshops, 2011, no. i, pp. 104–109.

[80] H. Shahriar, M. Zulkernine, MUTEC: Mutation-based Testing of Cross Site Scripting School of Computing, in: Software Engineering for Secure Systems, 2009. SESS '09. ICSE Workshop on, 2009, pp. 47–53.

[81] H. Shahriar, M. Zulkernine, S2XS2: A Server Side Approach to Automatically Detect XSS Attacks, in: 2011 IEEE Ninth International Conference on Dependable, Autonomic and Secure Computing, 2011, pp. 7–14.

[82] H. Shahriar, M. Zulkernine, Trustworthiness testing of phishing websites: a behavior model-based approach, Futur. Gener. Comput. Syst. 28 (8) (2012) 1258–1271. October.

[83] J. Shanmugam, M. Ponnavaikko, A solution to block Cross Site Scripting Vulnerabilities based on Service Oriented Architecture, in: 6th IEEE/ACIS Int. Conf. Comput. Inf. Sci. (ICIS 2007), no. Icis, 2007, pp. 861–866.

[84] J. Shanmugam, M. Ponnavaikko, Behavior-Based Anomaly Detection on the Server Side to Reduce the Effectiveness of Cross Site Scripting Vulnerabilities, Third Int. Conf. Semant. Knowl. Grid (SKG 2007), pp. 350–353, October 2007.

[85] J. Shanmugam, M. Ponnavaikko, XSS Application Worms: New Internet Infestation and Optimized Protective Measures, Eighth ACIS Int. Conf. Softw. Eng. Artif. Intell. Networking, Parallel/Distributed Comput. (SNPD 2007), July 2007, pp. 1164–1169.

[86] L.K. Shar, H.B.K. Tan, Auditing the defense against cross site scripting in web applications, in: 2010 International Conference on Security and Cryptography SECRYPT, 2010, pp. 1–7.

[87] L.K. Shar, H.B.K. Tan, Auditing the XSS defence features implemented in web application programs, IET Softw. 6 (4) (2012) 377.

[88] L.K. Shar, H.B.K. Tan, Automated removal of cross site scripting vulnerabilities in web applications, Inf. Softw. Technol. 54 (5) (2012) 467–478. May.

[89] L.K. Shar, H.B.K. Tan, Mining input sanitization patterns for predicting SQL injection and cross site scripting vulnerabilities, in Proceedings – 34th International Conference on Software Engineering, ICSE 2012, 2012, pp. 1293–1296.

[90] L.K. Shar, H.B.K. Tan, Predicting common web application vulnerabilities from input validation and sanitization code patterns, in: Proc. 27th IEEE/ACM Int. Conf. Autom. Softw. Eng. – ASE 2012, 2012, p. 310.

[91] P. Sharma, R. Johari, S.S. Sarma, Integrated approach to prevent SQL injection attack and reflected cross site scripting attack, Int. J. Syst. Assur. Eng. Manag. 3 (4) (2012) 343–351. September.

[92] K. Sivakumar, K. Garg, Constructing a 'Common Cross Site Scripting Vulnerabilities Enumeration (CXE)' using CWE and CVE, Lect. Notes Comput. Sci. 4812 (2007) 277–291.

[93] J. Somorovsky, M. Heiderich, R. Bochum, N. Gruschka, L. Lo Iacono, All Your Clouds are Belong to us – Security Analysis of Cloud Management Interfaces, in: CCSW '11: Proceedings of the 3rd ACM workshop on Cloud Computing Security Workshop, 2011, pp. 3–14.

[94] J. Stuckman, J. Purtilo, A Testbed for the Evaluation of Web Intrusion Prevention Systems, in: 2011 Third International Workshop on Security Measurements and Metrics, 2011, pp. 66–75.

[95] F. Sun, L. Xu, Z. Su, Client-side detection of XSS worms by monitoring payload propagation, Lect. Notes Comput. Sci. 5789 (2009) 539–554.

[96] Y. Sun, D. He, Model Checking for the Defense against Cross-Site Scripting Attacks, in: 2012 International Conference on Computer Science and Service System, 2012, pp. 2161–2164.

[97] S. Sundareswaran, A.C. Squicciarini, XSS-Dec : A hybrid solution to mitigate cross-site scripting attacks, Lect. Notes Comput. Sci. 7371 (2012) 223–238.

[98] M. Takesue, A Protection Scheme against the Attacks Deployed by Hiding the Violation of the Same Origin Policy, in: 2008 Second International Conference on Emerging Security Information Systems and Technologies, 2008, pp. 133–138.

[99] S. Tang, C. Grier, O. Aciicmez, S.T. King, Alhambra : A System for Creating, Enforcing, and Testing Browser Security Policies, in: WWW '10: Proceedings of the 19th international conference on World wide web, 2010, pp. 941–950.

[100] Z. Tang, H. Zhu, Z. Cao, S. Zhao, L-WMxD: Lexical based Webmail XSS Discoverer, in: 2011 IEEE Conference on Computer Communications Workshops INFOCOM WKSHPS, 2011, pp. 976–981.

[101] S. Shalini, S. Usha, Prevention Of cross-site scripting attacks (XSS) on web applications in the client side, Int. J. Comput. Sci. Issues 8 (4) (2011) 650–654.

[102] S. Tiwari, R. Bansal, D. Bansal, Optimized client side solution for cross site scripting, in: 2008 16th IEEE International Conference on Networks, 2008, pp. 1–4.

[103] D.-R.T.D.-R. Tsai, a Y. Chang, P.L.P. Liu, H.-C.C.H.-C. Chen, Optimum tuning of defense settings for common attacks on the web applications, in: 43rd Annual 2009 International Carnahan Conference on Security Technology, 2009, pp. 89–94.

Please cite this article in press as: I. Hydara et al., Current state of research on cross-site scripting (XSS) – A systematic literature review, Inform. Softw. Technol. (2014), http://dx.doi.org/10.1016/j.infsof.2014.07.010

[104] S.C.V., S. Selvakumar, Bixsan: Browser Independent XSS Sanitizer for prevention of XSS attacks, ACM SIGSOFT Softw. Eng. Notes, vol. 36, no. 5, p. 1, September 2011.

[105] M. Van Gundy, H. Chen, Noncespaces: using randomization to defeat cross-site scripting attacks, Comput. Secur. 31 (4) (2012) 612–628. June.

[106] S. Van-Acker, N. Nikiforakis, L. Desmet, W. Joosen, F. Piessens, FlashOver: Automated Discovery of Cross-site Scripting Vulnerabilities in Rich Internet Applications, in: ASIACCS '12: Proceedings of the 7th ACM Symposium on Information, Computer and Communications Security, 2012, pp. 12–13.

[107] V.N. Venkatakrishnan, P. Bisht, M. Ter Louw, M. Zhou, WebAppArmor: a framework for robust prevention of attacks on web applications (invited paper), Lect. Notes Comput. Sci. 6503 (2010) 3–26.

[108] P. Vogt, F. Nentwich, N. Jovanovic, E. Kirda, C. Kruegel, G. Vigna, Cross-Site Scripting Prevention with Dynamic Data Tainting and Static Analysis, in: 14th Annual Network and Distributed System Security Symposium Proceedings, NDSS Symposium 2007, 2007.

[109] S. Wang, Y. Chang, W. Chiang, W. Juang, Investigations in Cross-site Script on Web-systems Gathering Digital Evidence against Cyber-Intrusions, in: Future Generation Communication and Networking (FGCN 2007) (Volume: 2), 2007, pp. 125–129.

[110] Y. Wang, Z. Li, T. Guo, Program Slicing Stored XSS Bugs in Web Application, in: 2011 Fifth International Conference on Theoretical Aspects of Software Engineering, 2011, pp. 191–194.

[111] G. Wassermann, Z. Su, Static Detection of Cross-Site Scripting Vulnerabilities, in: ICSE '08: Proceedings of the 30th international conference on Software engineering, 2008, pp. 171–180.

[112] J. Weinberger, P. Saxena, D. Akhawe, M. Finifter, R. Shin, D. Song, A Systematic Analysis of XSS Sanitization in Web Application Frameworks, in: ESORICS'11: Proceedings of the 16th European Conference on Research in Computer Security, 2011, pp. 150–171.

[113] P. Wurzinger, C. Platzer, C. Ludl, E. Kirda, C. Kruegel, SWAP: Mitigating XSS Attacks using a Reverse Proxy, in: Software Engineering for Secure Systems, 2009. SESS '09. ICSE Workshop on, 2009, pp. 33–39.

[114] Z. Xin-hua, W. Zhi-jian, A Static Analysis Tool for Detecting Web Application Injection Vulnerabilities for ASP Program, in: 2nd International Conference on e-Business and Information Security (EBISS), 2010, pp. 1–5.

[115] P. Xiong, B. Stepien, L. Peyton, Model-based penetration test framework for web applications using TTCN-3, Lect. Notes Bus. Inf. Process. 26 (2009) 141–154.

[116] F. Yu, M. Alkhalaf, T. Bultan, STRANGER: An automata-based string analysis tool for PHP, Lect. Notes Comput. Sci. 6015 (2010) 154–157.

[117] F. Yu, T. Bultan, B. Hardekopf, String abstractions for string verification, Lect. Notes Comput. Sci. 6823 (2011) 20–37.

[118] L.Z.L. Zhang, Q.G.Q. Gu, S.P.S. Peng, X.C.X. Chen, H.Z.H. Zhao, D.C.D. Chen, D-WAV: A Web Application Vulnerabilities Detection Tool Using Characteristics of Web Forms, in: Software Engineering Advances ICSEA 2010 Fifth International Conference on, 2010, pp. 501–507.

[119] Q. Zhang, H. Chen, J. Sun, An execution-flow based method for detecting Cross-site Scripting attacks, in: Software Engineering and Data Mining SEDM 2010 2nd International Conference on, 2010, pp. 160–165.

[120] Q. Zhenyu, X. Jing, L. Baoguo, T. Fang, MBDS: Model-Based Detection System for Cross Site Scripting, in: IET Conference on Wireless, Mobile and Sensor Networks, 2007, 2007, pp. 849–852.

[121] J.Y. Halpern, D.C. Parkes, Journals for certification, conferences for rapid dissemination, Commun. ACM 54 (8) (2011) 36–38.

[122] L. Fortnow, Time for computer science to grow up, Commun. ACM 52 (8) (2009) 33–35.

[123] M. Franceschet, The role of conference publications in CS, Commun. ACM 53 (12) (2010) 129–132.

[124] R. Barnett, XSS Street-Fight: The Only Rule Is There Are No Rules, 2011.

[125] D. Zimmer, "Real World XSS," *XSSed.com*, 2008 [Online]. <http://www.xssed.com/article/21/Paper_Real_World_XSS/>.

[126] Veracode Inc., How to Prevent Cross-site Scripting Attacks: Expert Tactics, 2011.

[127] J. Rafail, Cross-Site Scripting Vulnerabilities, 2001.

[128] Rapid7, Web Application Security – Managing Cross-Site Scripting, The Number One Item on OWASP's Top Ten List.

[129] W. Paper, Detecting Persistent Cross-site Scripting, pp. 1–12.

[130] OWASP, XSS (Cross Site Scripting) Prevention Cheat Sheet, *OWASP* [Online]. <https://www.owasp.org/index.php/XSS_(Cross_Site_Scripting)_Prevention_Cheat_Sheet>.